

TDP007 - Dugga 1

2025-02-04

Regler

- All kod efterrättas under denna examination
- Inga elektroniska hjälpmedel får medtas. Mobiltelefon ska vara avstängd och ligga i jacka eller väska.
- Inga ytterkläder eller väskor vid skrivplatsen.
- Student får lämna salen tidigast en timme efter start.
- Vid toalettbesök ska vakt informeras.
- All form av kontakt mellan studenter under tentamens gång är strängt förbjuden.
- Böcker och anteckningssidor kan komma att granskas av assistent, vakt eller examinator under tentamens gång.
- Frågor om specifika uppgifter eller om tentamen i stort ska ställas via tentasystemets kommunikationsklient.
- Systemfrågor kan ställas till assistent i sal genom att räcka upp handen.
- Endast uppgifter inskickade före tentamenstidens slut rättas.

Hjälpmedel	En Rubybok (exempelvis the pickaxe) Ett A4-ark med egna anteckningar Tillgång till webresurser: ruby-docs, rubular och tidig version av kursboken
------------	--

Information

Betygssättning vid tentamen

Tentamen består av 3 delar. Del 1 är uppgifter på grund-nivå som relaterar till tekniker från kursens första två seminarier, dessa bedöms som G/U. Del 2 är uppgifter på grund-nivå som relaterar till tekniker från kursens andra två seminarier, dessa bedöms som G/U. Del 3 består av mer avancerade uppgifter, där varje uppgift relaterar till någon teknik från kursens fyra seminarier. Uppgifter på del 3 ger upp till 10 poäng. Betyg på tentamen ges enligt följande tabell. För alla betyg så måste förutom de insamlade poängen alla uppgifter på Del 1 och Del 2 lösas med godkänt betyg.

Insamlade poäng Del 3	Betyg
10	3
17	4
24	5

Duggor

Den första duggan består av två delar. Del 1 är uppgifter på grundnivå som relaterar till tekniker från kursens första två seminarier och bedöms som G/U. Del 2 består av en mer avancerad XML-uppgift som ger upp till 10 poäng. Om alla uppgifter på Del 1 har godkänt betyg från denna dugga så har du Del 1 på ordinarie tentamen tillgodo.

Den andra duggan består av två delar. Del 1 är uppgifter på grundnivå som relaterar till tekniker från kursens sista två seminarier och bedöms som G/U. Del 2 består av en mer avancerad parser-uppgift som ger upp till 10 poäng. Om alla uppgifter på Del 1 har godkänt betyg från denna dugga så har du Del 2 på ordinarie tentamen tillgodo.

Om del 1 är avklarad på båda duggorna och du har samlat in totalt 14 poäng från duggorna så har du förutom Del 1 och Del 2 på ordinarie tentamen tillgodo också betyg 3 på tentamensmomentet.

Inloggning

Logga in på datorn i labbsalen med ditt liu-id och lösenord. Detta är samma inloggningsuppgifter som du använder i Lisam.

Skrivbordsmiljön

När du kommit in i tentasystemet har du en normal skrivbordsmiljö (Mate-session i Ubuntu). Efter en stund kommer även din kommunikationsklient att dyka upp automatiskt. Startmenyn är nedskalad till enbart det som examinator bedömt relevant. Andra program kan fortfarande finnas tillgängliga genom att starta dem från en terminal. Observera att en del program som använder nätverkstjänster inte fungerar normalt, eftersom nätverket inte är åtkomligt.

När du är inloggad är det viktigt att du har tentaklienten igång hela tiden. Om den inte dykt upp fem minuter efter inloggning och inte heller när du startar den manuellt från menyn (fisken) tar du kontakt med assistent eller vakt i sal.

Terminalkommandon

`ruby` används för att köra en ruby-fil.

`irb` används för att starta ruby i interaktivt läge.

`ri` används för att komma åt den inbyggda dokumentationen.

Ruby-docs

På tentan har du tillgång till referenssidorna på <https://ruby-doc.org/> (både core och std-lib finns tillgänglig) via webbläsaren Chrome. Starta `chromium-browser` i terminalen eller välj lämpligt alternativ från startmenyn. Observera att allt utom referenssidorna är avstängt. Om du inte kan komma åt en sida du tycker hör till referenssidorna (som kanske blockerats av misstag) kan du skicka ett meddelande via tentaklienten. Tag hjälp av assistent i sal om det inte fungerar.

Rubular

På tentan har du tillgång till sidan <https://rubular.com/>.

Givna filer

Eventuella givna filer finns i katalogen `given_files`. Denna underkatalog är skrivskyddad, så det är ingen risk du råkar ändra på dessa filer. Skrivskyddet gör dock att du måste kopiera in de givna filer du vill använda till tentakontots hemkatalog. Hur du listar och kopierar filer ska du kunna. Hemkatalogen står du i från början, och du kommer alltid tillbaka till den genom att bara exekvera kommandot `cd` i terminalen.

Avslutning

När du skickat in alla uppgifter och känner dig färdig kan du logga ut och lämna salen. Antalet poäng meddelas i efterhand via webreg.

Avsluta alla öppna program och logga ut ur datorn. Lämna inte datorn innan du ser att du är utloggad.

Del 1: Uppgift 1 - Regexp

I den givna filen `uppgift1.rb` finns det en sträng som innehåller fyra giltiga mobilnummer. Din uppgift är att skapa ett reguljärt uttryck som kan matcha alla dessa telefonnummer och ersätta dem med en annan sträng, t.ex. "XXXXXXXXXX" för att anonymisera dem. Exakt hur du anonymiserar alla telefonnummer du matchar spelar ingen roll. Du kan utgå från att alla telefonnummer har följande struktur: De har inga mellanslag eller bindestreck i sig, och de börjar antingen på 0 eller +46. Efter 0 eller +46 är det alltid 9 stycken siffror till, och den första siffran ska alltid vara 7.

Funktionella krav:

```
XXXXXXXXXX
XXXXXXXXXX
01101000 01100101 01101100 01101100 01101111
XXXXXXXXXX
+46831947266
0801234567
+47891684858
XXXXXXXXXX
```

Del 1: Uppgift 2 - SAX-parser

I filen `books.xml` hittar du ett antal sci-fi-böcker, några av dem är en del av en bokserie. Din uppgift är att skriva ut namnet på alla böcker som är en del av en serie (från elementet `<series>`). Du ska skriva ut namnet på varje bok som är med i en serie så att det tydligt framgår vilken serie boken tillhör. Det är okej att lagra serien och böckerna i en valfri datastruktur och skriva ut denna, så länge resultatet är rätt. Observera att utskriften i de funktionella kraven bara är ett exempel på hur utskriften kan se ut, din utskrift kan exempelvis ha en rad per bok.

Funktionella krav:

```
{
  "Dune"=>["Dune", "Dune Messiah"],
  "Bobiverse"=>["We Are Legion (We Are Bob)", "For We Are Many"]
}
```

Del 1: Uppgift 3 - DOM-parser

Utgå från den givna filen `staff.xml`. Använd en dom-parser för att skriva ut namnen på alla personer som jobbar inom teamen `SEO` och `Software Development`. Din lösning skall fungera även om antalet element eller innehållet i element (men inte strukturen) i xml-filen ändras över tid.

Funktionella krav:

```
Jane Doe
John Smith
Emily Davis
Laura Martin
Tom Anderson
```

Del 2: Uppgift 4 - DOM-parser

Utgå ifrån filen `staff.xml`. Du ska med hjälp av en dom-parser läsa ut information om alla personer som arbetar i divisionen marketing. Skapa en enkel klass som representerar en person, klassen ska spara vad personen heter, vilket team de arbetar i och deras erfarenhet. Använd sedan dom-parsern för att läsa ut den informationen om alla som jobbar i divisionen marketing och skapa instanser av klassen för varje person som jobbar där. Spara varje person du skapar i en lämplig behållare och skriv tillsist ut behållaren (se funktionella krav för formattering). Din lösning skall fungera även om antalet element (men inte strukturen) i xml-filen ändras över tid.

Funktionella krav:

```
Anna Taylor, has worked in Content Creation for 6 years
Robert Johnson, has worked in Content Creation for 4 years
Laura Martin, has worked in SEO for 5 years
Tom Anderson, has worked in SEO for 3 years
```